

AMENDMENTS TO THE CLAIMS

Prior to this amendment, claims 1, 2, 4, 13, 14, and 16 were pending in the subject application. Claim 1 is currently amended, as set forth herein. All claims currently pending and under consideration in the referenced application are shown below. This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Currently Amended) In a threaded computing environment having a plurality of contexts, each context capable of containing a queue, context settings, a context dictionary, and objects, a method for allocating the access of threads to a user interface context, the method comprising:

receiving a request to access the user interface context from a first thread;

determining whether the user interface context is presently being accessed
by a second thread, and

if the user interface context is presently being accessed by a second thread,
denying the request to access the user interface context received from the first
thread;

if the user interface context is not presently being accessed by a second
thread, allowing the request to access the user interface context received from the
first thread;

maintaining thread settings associated with threads;

maintaining context settings in the user interface context; and

applying the context settings and the context dictionary of the user interface context in place of the thread settings of any thread accessing the user interface context.

2. (Original) The method for allocating the access of threads to user interface context of claim 1, the method further comprising:

maintaining a context record associated with each thread that identifies the contexts accessed by the thread, the most recent entry in the context record indicating the context presently being accessed by the thread;

when a thread accesses an object in the user interface context, checking the most recent entry in the context record associated with the thread;

determining whether the most recent entry in the context record matches the context of the object being accessed; and

if the most recent entry in the context record does not match the context of the object being accessed, raising an exception.

3. (Canceled).

4. (Previously Presented) The method for allocating the access of threads to a user interface context of claim 13, the method further comprising restoring the thread settings when a thread departs the user interface context.

5-12. (Canceled).

13. (Previously Presented) One or more computer-storage media having computer-executable instructions embodied thereon that, when executed, perform a method for allocating the access of threads to a user interface context in a threaded computing environment having a plurality of contexts, each context capable of containing a queue, context settings, a context dictionary, and objects, the method for allocating the access of threads to a user interface context comprising:

receiving a request to access the user interface context from a first thread, wherein the user interface context comprises one or more objects;

determining whether the user interface context is presently being accessed by a second thread, and:

if the user interface context is presently being accessed by a second thread, denying the request to access the user interface context received from the first thread;

if the user interface context is not presently being accessed by a second thread, allowing the request to access the user interface context received from the first thread;

maintaining thread settings associated with threads;

maintaining context settings in the user interface context;

maintaining context dictionary in the user interface context, wherein the context dictionary comprises information from one or more sources; and

applying the context settings and the context dictionary of the user interface context in place of the thread settings of any thread accessing the user interface context.

14. (Previously Presented) The one or more computer-storage media of claim 13, the method for allocating the access of threads to a user interface further comprising:

maintaining a context record associated with each thread that identifies the contexts accessed by the thread, the most recent entry in the context record indicating the context presently being accessed by the thread;

when a thread accesses one of the one or more objects in the user interface context, checking the most recent entry in the context record associated with the thread;

determining whether the most recent entry in the context record matches the context of the object being accessed; and

if the most recent entry in the context record does not match the context of the one of the one or more objects being accessed, raising an exception.

15. (Canceled).

16. (Previously Presented) The one or more computer-storage media of claim 13, the method for allocating the access of threads to a user interface further comprising restoring the thread settings when a thread departs the user interface context.

17-24. (Canceled).